Версионирование данных. Теория и практика

Максим Тремпольцев

Data Warehouse (DWH)

Термин появился в 1988 году после публикации **Барри Девлином** (Barry Devlin) и **Полом Мерфи** (Paul Murphy) статьи в **IBM Systems Journal** с описанием архитектурной модели для потоков данных из различных источников в систему поддержки принятия решений (Decision Support System, DSS).

Что такое DWH?

Хранилище данных служит для централизации и консолидации больших объемов **исторических** данных из различных источников.

https://www.oracle.com/ru/database/what-is-a-data-warehouse/

Data sources | **Data warehouse** | **Data marts** | **Users**

Operational system

ERP

CRM

Plain files

ETL

Extract, Transform, Load

miro

От ядра инвестиционного бизнеса Альфа-Банка…

https://www.youtube.com/watch?v=PzEMpGOz7eg

...до Tarantool Data Grid

https://www.tarantool.io/ru/datagrid/

Slowly Changing Dimension (SCD)

Термин введен Ральфом Кимбелом (Ralph Kimball) в 1996 году в книге «The Data Warehouse Toolkit».

Что такое SCD?

Механизм отслеживания изменений в данных.

* Применяется, если данные меняются не очень часто и не по расписанию.

Типы SCD

- Упоминается 8 типов
- Базовых типов 5
- Остальные типы являются комбинацией трех базовых типов
- Нет самого правильного/лучшего типа

SCD Type 0

- Оригинал не изменяется
- Как правило, применяется для хранения фактов, информации о событиях и т.д.

SCD Type 1

Хранится актуальное значение

id	name	email
de4c8987	Bobby Brown	bb@gmail.com
fc05b042	Jessica Gray	jg@gmail.com

id	name	email
de4c8987	Bobby Brown	bb@gmail.com
fc05b042	Jessica Gray	jgray@mail.com

SCD Type 1

```lua
local format = {
    { name = 'id',      type = 'string' },
    { name = 'name',    type = 'string' },
    { name = 'email',   type = 'string' }
}

space = box.schema.space.create(name)
space:format(format)
space:create_index('id', { parts = { 'id' }})
```

SCD Type 1

```lua
local function get_last(self, id)
    return self.space:get({ id })
end

local function insert(self, profile)
    self.space:replace(flatten(profile))
end

local function update(self, id, updates)
    self.space:update(id, updates)
end
```

SCD Type 1

+ Нет избыточности
+ Простота
- Не сохраняется история

SCD Type 2

Добавляется новая запись

id	version	name	email
de4c8987	01.05.2021	Bobby Brown	bb@gmail.com
fc05b042	05.05.2021	Jessica Gray	jg@gmail.com
fc05b042	17.05.2021	Jessica Gray	jgray@mail.com

SCD Type 2

```
local format = {
    { name = 'id',      type = 'string' },
    { name = 'version', type = 'unsigned' },
    { name = 'name',    type = 'string' },
    { name = 'email',   type = 'string' }
}

space = box.schema.space.create(name)
space:format(format)
space:create_index('id', { parts = { 'id', 'version' }})
```

SCD Type 2

```lua
local function get_last(self, id)
    return self.space.index.id:max({ id })
end

local function get_all(self, id)
    local res = {}
    for _, v in self.space:pairs({ id }, 'REQ') do
        table.insert(res, v)
    end
    return res
end
```

SCD Type 2

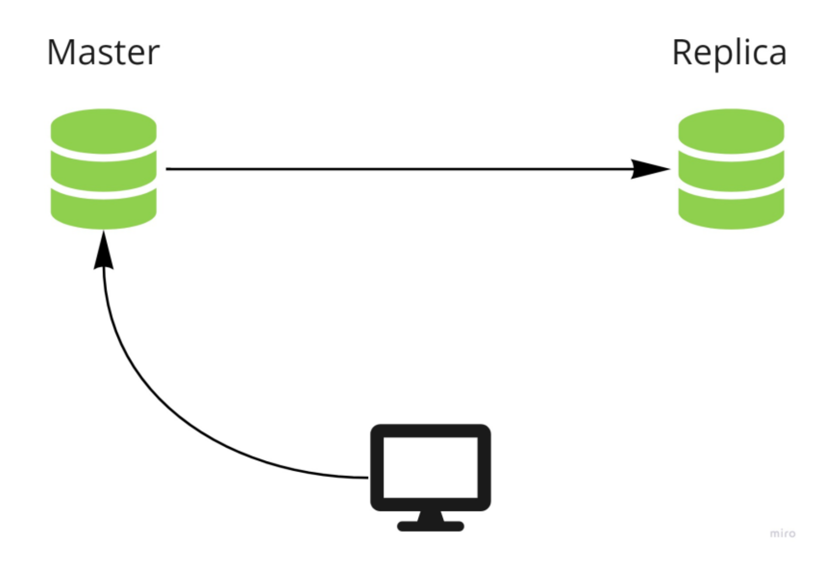
```lua
local function insert(self, profile)
    profile.version = clock.time64()
    self.space:insert(flatten(profile))
end

local function update(self, id, updates)
    table.insert(updates, { '=', 'version', clock.time64() })
    local t = get_last(self, id)
    self.space:insert(t:update(updates))
end
```

Почему версия – это время?

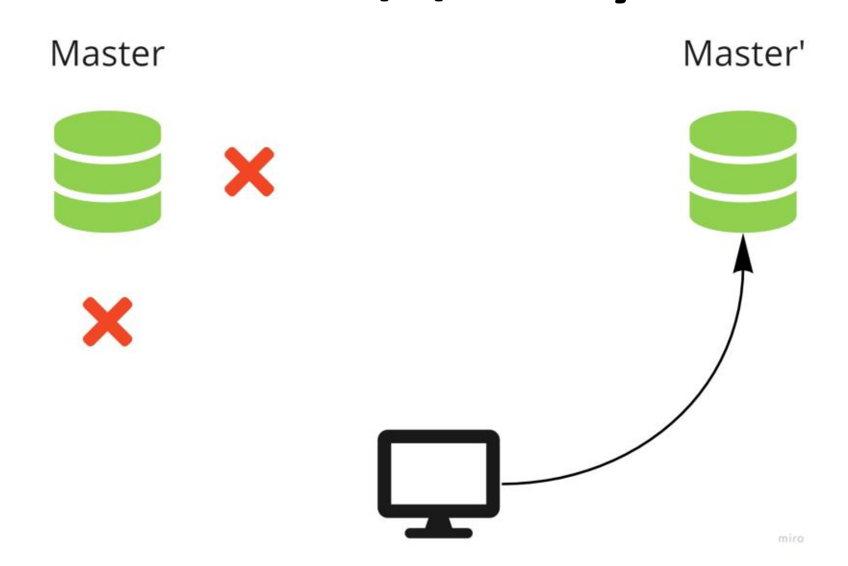
- Исторические срезы: можно посмотреть, как выглядела БД в определенное время
- Устойчиво к конфликтам репликации

Нормальная работа системы



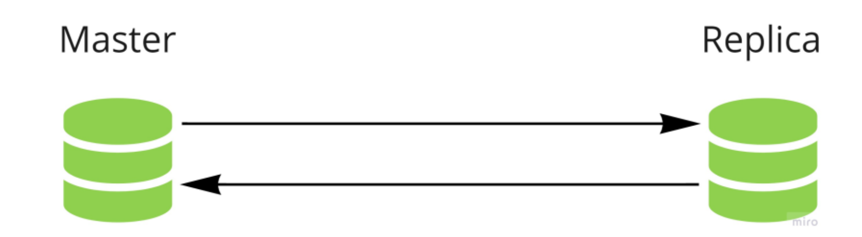
- Данные идут на мастер
- От мастера данные дублируются на реплику

Мастер стал недоступен



- Выбран новый мастер
- Неудавшиеся запросы будут перенаправлены на новый мастер

Старый мастер стал доступен



- Происходит обмен данными до состояния, когда наборы данных на мастере и реплике будут идентичны
- Если на мастере и реплике окажется запись с одним и тем же первичным ключом, то это приведет к конфликту репликации

SCD Type 2

+ Полная история
+ Прост в реализации
- Есть избыточность
- Страдает производительность аналитических запросов из-за необходимости определения актуальной версии

SCD Type 2

Добавляется новая запись

id	version	name	email	current
de4c8987	01.05.2021	Bobby Brown	bb@gmail.com	true
fc05b042	05.05.2021	Jessica Gray	jg@gmail.com	false
fc05b042	17.05.2021	Jessica Gray	jgray@mail.com	true

SCD Type 2

+ Полная история
- Есть избыточность
- Страдает производительность обновления данных из-за необходимости обновлять предыдущую запись

SCD Type 2

Добавляется новая запись

id	start	end	name	email
de4c8987	01.05.2021	null	Bobby Brown	bb@gmail.com
fc05b042	05.05.2021	17.05.2021	Jessica Gray	jg@gmail.com
fc05b042	17.05.2021	null	Jessica Gray	jgray@mail.com

SCD Type 2

+ Полная история
- Есть избыточность
- Страдает производительность обновления данных из-за необходимости обновлять предыдущую запись

SCD Type 2

Добавляется новая запись

id	start	end	name	email	current
de4c8987	01.05.21	null	Bobby Brown	bb@gmail.com	true
fc05b042	05.05.21	17.05.21	Jessica Gray	jg@gmail.com	false
fc05b042	17.05.21	null	Jessica Gray	jgray@mail.com	true

SCD Type 2

+ Полная история
+ Можно завершать жизненный цикл серии изменений без удаления
- Есть избыточность
- Страдает производительность обновления данных из-за необходимости обновлять предыдущую запись

SCD Type 3

Для изменяемых значений добавляется новая колонка

id	name	old_email	email
de4c8987	Bobby Brown	bb@gmail.com	bb@gmail.com
fc05b042	Jessica Gray	null	jg@gmail.com

id	name	old_email	email
de4c8987	Bobby Brown	bb@gmail.com	bb@gmail.com
fc05b042	Jessica Gray	jg@gmail.com	jgray@mail.com

SCD Type 3

```lua
local format = {
    { name = 'id',        type = 'string' },
    { name = 'name',      type = 'string' },
    { name = 'old_email', type = 'string', is_nullable = true },
    { name = 'email',     type = 'string' }
}

space = box.schema.space.create(name)
space:format(format)
space:create_index('id', { parts = { 'id' }})
```

SCD Type 3

```lua
local function clear_old(t)
    return t:update({{ '=', 'old_email', box.NULL }})
end

local function get_last(self, id)
    local t = self.space:get({ id })
    return clear_old(t)
end

local function get_all(self, id)
    local t = self.space:get({ id })
    if t.old_email == nil then return { t } end

    local old = t:update({{ '=', 'email', t.old_email }})

    return { clear_old(t), clear_old(old) }
end
```

SCD Type 3

```lua
local function insert(self, profile)
    self.space:insert(flatten(profile))
end

local function update(self, id, updates)
    local t = get_last(self, id)

    for _, u in ipairs(updates) do
        if u[2] == 'email' then
            table.insert(updates, { '=', 'old_email', t.email })
        end
    end

    self.space:replace(t:update(updates))
end
```

SCD Type 3

\+ Не хранятся неизмененные значения

\- Ограниченная история

* Хорошо работает вместе с 4 типом, если требуется быстро определять предыдущее значение, например, предыдущий статус задачи

SCD Type 4

Добавляется таблица с историей

id	version	name	email
de4c8987	01.05.2021	Bobby Brown	bb@gmail.com
fc05b042	17.05.2021	Jessica Gray	jgray@mail.com

Историческая таблица

id	version	name	email
fc05b042	05.05.2021	Jessica Gray	jg@gmail.com

SCD Type 4

```
local format = {
    { name = 'id',        type = 'string' },
    { name = 'version',   type = 'unsigned' },
    { name = 'name',      type = 'string' },
    { name = 'email',     type = 'string' }
}

space = box.schema.space.create(name)
space:format(format)
space:create_index('id', { parts = { 'id' }})

history_space = box.schema.space.create(name .. "_history")
history_space:format(format)
history_space:create_index('id', { parts = { 'id', 'version' }})
```

SCD Type 4

```lua
local function get_last(self, id)
    return self.space:get({ id })
end

local function get_all(self, id)
    local res = { get_last(self, id) }
    for _, v in self.history_space:pairs({ id }, 'REQ') do
        table.insert(res, v)
    end
    return res
end
```

SCD Type 4

```lua
local function insert(self, profile)
    self.space:insert(flatten(profile))
end

local function update(self, id, updates)
    local t = get_last(self, id)
    self.space:replace(t:update(updates))
end
```

SCD Type 4

```lua
space:before_replace(function(old, new)
    if old ~= nil then
        history_space:insert(old)
    end

    return new:update({{ '=', version_field, clock.time64() }})
end)
```

SCD Type 4

+ Полная история
+ Быстрый поиск по актуальным данным
+ Возможно хранение истории только обновляемых полей
+ Возможно хранение истории в холодном хранилище
- Обновление требует записи в две таблицы
- Присутствует избыточность

Базовые типы

Type 0	Оригинал не изменяется
Type 1	Хранятся актуальные значения
Type 2	Добавляется новая запись
Type 3	Добавляется новая колонка
Type 4	Добавляется историческая таблица

Все остальные типы являются комбинацией базовых

SCD Type 5, 6, 7, …

- I'm trying to understand how SCD Type 5,6 & 7 work. … … … However, I'm still unable to understand how type 5 & 7 work and when to use them.
- I wouldn't worry too much - all the types above Type 3 have been called Type 6 at various times. Basically there are a range of techniques to deal with more complex history tracking, and it is up to you to pick the mix that works for your situation.

https://stackoverflow.com/questions/43197559/understanding-slowly-changing-dimension-scd-type-5-and-7-with-examples

SCD Type 5 (1 + 4)

- Дополнительная таблица для быстро изменяющихся атрибутов, как в типе 4
- Ключ, ссылающийся на дополнительную таблицу, перезаписывается, как в типе 1

* Из соображений производительности внешняя таблица может быть встроена в базовую

SCD Type 6 (1 + 2 + 3)

- Добавляется новая запись (Type 2)
- Добавляется колонка для актуального значения (Type 3)
- При обновлении записи обновляется поле у всех предыдущих записей

id	version	name	history_address	current_address
de4c8987	01.05.21	Bobby Brown	Benin, Wadeside	Benin, Wadeside
fc05b042	05.05.21	Jessica Gray	Samoa, Bushport	Tonga, East Joy
fc05b042	17.05.21	Jessica Gray	Tonga, East Joy	Tonga, East Joy

SCD Type 6

+ Полная история
+ Получение актуального значения эффективно
- Обновление требует обновления всех предыдущих версий
- Присутствует избыточность

SCD Type 7

Альтернатива типу 6

id	version	name	address
de4c8987	01.05.21	Bobby Brown	Benin, Wadeside
fc05b042	05.05.21	Jessica Gray	Samoa, Bushport
fc05b042	17.05.21	Jessica Gray	Tonga, East Joy

id	current_address
de4c8987	Benin, Wadeside
fc05b042	Tonga, East Joy

SCD Type 6

+ Полная история
+ При обновлении не требуется изменять все предыдущие версии
- Обновление требует записи в две таблицы
- Присутствует избыточность

Спасибо!

Максим Тремпольцев
mt@devexp.ru

Код примеров доступен на:
https://github.com/mtrempoltsev/highload2021_versioning